Claims 7, 9, 11, 15, 18-20, 22, 26, 27, and 29-41 are pending in the application. Claims 7, 9, 11, 15, 18-20, 22, 26-27 and 29-41 stand rejected under 35 USC § 103(a) as being unpatentable over U.S. Patent No. 6,125,391 to Meltzer et al. in view of U.S. Patent No. 6,446,077 to Straube et al. and U.S. Patent No. 6,772,396 to Cronin.

Reconsideration is requested. The rejections are traversed. No new matter is added. Claim 42 is added. Claims 7, 9, 11, 15, 18-20, 22, 26, 27, and 29-42 remain in the case for consideration.

REJECTIONS UNDER 35 U.S.C. § 103(a)

Claim 7 is directed toward a software program for facilitating the use of a distributed directory running in a computer network, the program comprising being stored on a recordable medium and including instructions for: receiving an event from the distributed directory into an XML generator, the distributed directory including a reference to at least one resource on one of at least two computers on the computer network, the event representing a change to the distributed directory; converting the event into XML data representing the event; transforming the XML data representing the event to a first predetermined format by a transformation processor using a first stylesheet, the first predetermined format being responsive to a first application running in the computer network; transmitting the transformed XML data representing the event to the first application; transforming the XML data representing the event to a second predetermined format by the transformation processor using a second stylesheet, the second predetermined format being responsive to a second application running in the computer network; and transmitting the transformed XML data representing the event to the second application.

Claim 15 is directed toward a software program for facilitating the use of a distributed directory running in a computer network, the program comprising instructions for: receiving a first event from a first application in a first native application format, the first event representing a first change to the distributed directory; converting the first event into markup language data; transforming the first event to a predetermined format by a transformation processor using a transformation profile, the predetermined format being responsive to the distributed directory, the transformation profile including formatting instructions for transforming the markup

language data to the predetermined format, the distributed directory including a reference to at least one resource on one of at least two computers on the computer network; transmitting the transformed first event to the distributed directory; receiving a second event from a second application in a second native application format, the second event representing a second change to the distributed directory; converting the second event into markup language data; transforming the second event to the predetermined format by the transformation processor using the transformation profile; and transmitting the transformed second event to the distributed directory.

Claim 18 is directed toward a distributed computer system comprising: a first processor connected to a network for executing computer code; a second processor connected to the network for executing computer code; a first memory connected to the first processor; a second memory connected to the second processor; a distributed directory, wherein first and second portions of the distributed directory are stored in the first memory and the second memory, respectively; a first application, a portion of which being stored in one of the first memory and the second memory; a second application, a portion of which being stored in one of the first memory and the second memory; a first transformation profile defining a first predetermined format for use by the first application; a second transformation profile defining a second predetermined format for use by the second application; software for detecting a directory event in the distributed directory, the directory event representing a change to the distributed directory; software for transforming the directory event to the first predetermined format by using a generic transformation tool and the first transformation profile; software for transforming the directory event to the second predetermined format by using the generic transformation tool and the second transformation profile; software for providing to the first application the directory event transformed to the first predetermined format; and software for providing to the second application the directory event transformed to the second predetermined format.

Claim 32 is directed toward a method for interfacing with a distributed directory in a computing system, comprising: providing a first transformation profile defining a first predetermined format for use by a first application; providing a second transformation profile defining a second predetermined format for use by a second application; detecting an event in the distributed directory, the distributed directory including a reference to at least one resource on one of at least two computers on the computer network, the event representing a change to the distributed directory; transforming the event to the first predetermined format by using a

transformation tool and the first transformation profile; transforming the event to the second predetermined format by using the transformation tool and the second transformation profile; providing to the first application the event transformed to the first predetermined format; and providing to the second application the event transformed to the second predetermined format.

Claim 40 is directed toward a driver infrastructure for interfacing a distributed directory and applications comprising: a generator to receive a directory event from the distributed directory and to generate a generic data for the directory event, the distributed directory including a reference to at least one resource on one of at least two computers on a computer network, the directory event representing a change to the distributed directory; a first transformation profile defining a first predetermined format for use by a first application; a second transformation profile defining a second predetermined format for use by a second application; a transformation processor to transform the generic data for the directory event into a first application data for the first application using the first transformation profile and to transform the generic data for the directory event into a second application data for the second application using the second transformation profile; and a transmitter to transmit the first application data to the first application and to transmit the second application data to the second application.

Claim 42 is directed toward a distributed computer system according to claim 18, wherein the first portion of the distributed directory is different from the second portion of the distributed directory.

In response to the Appeal Brief filed October 5, 2005, the Examiner has reopened prosecution and added Straube as a new referenee. Straube teaches a system for propagating changes in a replicated database. When a change is made to an object in the database, the change is propagated to other copies of the database. For changes to inheritable information, the directory service agent identifies which object has been changed in a propagation queue. The propagator accesses the queue, finds all child objects affected by the change, and modifies their inherited information.

The Examiner argues that "[i]t would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Meltzer, Cronin, and Straube because it provides a method to sending information to the target applications in their own formats (Cronin reference, abstract) and apply the well known technique to new area, distributed

directory so the information could be exchanged between applications and directories" (Office Action dated January 24, 2006, page 4).

There are several problems with the Examiner's arguments. First, with respect to Meltzer and Cronin, the Examiner's arguments are the same as in previous Office Actions. The Applicant's renews all arguments previously made with respect to these references. For example, the Applicant has previously argued that Meltzer does not teach data flowing in two directions, as claimed in claim 15: Meltzer teaches data flowing in only one direction. These arguments are renewed herein.

Straube does not teach a distributed directory

The Examiner has cited to column 5, lines 11-31 of Straube as teaching a distributed directory. In fact, nowhere does Straube use this term. All that Straube says there about storage is that "[a] directory tree [which is one way to represent a database] is a visual representation of the relationships between multiple objects on a storage domain. The storage domain represents storage on a single computer system, or multiple computer systems having some form of shared storage" (column 5, lines 11-17). That Straube says "shared storage" is particularly significant. Absent any indication to the contrary (of which Straube provides none), the Applicant believes the term should be interpreted as "storage that is shared": in other words, a single storage that can be accessed by multiple computer systems. But if there is all storage is on a single storage device, then Straube fails to meet the definition of "distributed directory" that is provided in the claims: namely, a directory capable of storing references to resources on at least two computers.

Straube does not require data transformation

Straube only teaches communication of events with other copies of the database. This is made clear in column 6, lines 8-10, where Straube says that "this process of sending the message or replicating the change to other copies of the database is all that is needed". Note that Straube says "to other copies of the databsase": in other words, Straube is not transmitting data to applications.

This fact is important, because if Straube is only sharing information with other copies of the database, then the format used for making the change to the local copy of the database can be used with the copies. For example, the data received by the API on the local machine is the

same data needed to make the change on other copies of the data. In other words, Straube can just transmit the data used by the local API to the other copies of the database to propagate the changes.

This means that the Examiner's statement that there is a motivation to combine the references is incorrect: there is no motivation to combine Straube with either Meltzer or Cronin. The Examiner has recited that "Meltzer teaches . . . receiving an event from a server [and] . . . transforming . . . data representing the event to a first predetermined format . . . responsive to a first application running in the computer network" (Office Action dated January 24, 2006, page 3). In other words, according to Meltzer, the reason for data transformation is because the data generated by the server is not in a format usable by the first application. Assuming that the database of Straube could be analogized to both the server and the first application (a point the Applicant disputes: see below), then the two copies of the database both can process data in the same format. Thus, to use an intermediate format such as XML in Straube is unnecessary. But if data transformation is unnecessary, then there is no motivation to make such a transformation in Straube, which means that Straube cannot be combined with Meltzer as suggested by the Examiner.

Straube's database cannot be analogous to both the distributed directory and the first application

Recall that a problem with the prior the claimed invention addresses is that some applications are not designed to interact with the distributed directory. Accordingly, such applications store internal representations of the distributed directory, and operate on the local copies (see specification, page 2, lines 20-20). Because these applications store the internal representations in application-specific data formats, before these applications can process data from the distributed directory, the data needs to be transformed into a format these applications can receive (see specification, page 10, lines 7-22).

The Examiner says that Straube teaches "sending a message regarding the change in the distributed directory to other systems", citing to column 7, lines 2-7. This is an overstatement of Straube. Straube only describes propagating changes "to other copies of the database". This is important, because if Straube is only propagating changes to other copies of the database, Straube does not need to transform the data in any way. All copies of the database inherently use

the same format, because otherwise Straube would make specific mention of the need to change the data format. But if the data is being shared only with other copies of the database, there is no need for data transformation as suggested by Meltzer. To make the proposed combination would require taking Straube beyond its stated purpose, and as such there is no motivation to combine Straube and Meltzer as the Examiner suggests.

It is also worth noting that a database is not an application. A database is a storage place for data. Unless something acts on the database, the database itself does nothing. So it is up to an application to access the data and do something with it. The Examiner appears to have recognized this distinction, in that the Examiner analogizes the first application of the claims to "commercial functions 305, database functions 306 etc." of Meltzer, cited as column 23 line 64 through column 24, line 53. The Applicant specifically notes the use of the term "database functions" in the Examiner's citation: it is not the database itself, but rather a function that can use the database to which the Examiner analogizes the first application of the claims.

This point is important because if there is a function that is using the database, then the data is not actually being transformed into a predetermined format of the application. In fact, an application that uses a database such as Straube has no predetermined format of its own. The database defines the format. Any application that wants to utilize the data in the database need to conform to the requirements of the database for data access, and any application that conforms to the requirements of the database can access the database. This means that the format of the database is independent of any applications that access the database.

Further, there is no guarantee that any data in the database will necessarily ever be accessed by any function on the database. In other words, the data in the database of Straube is not guaranteed to ever be received by any application. In contrast, the claims of this patent application recite that the data is transmitted to the application, not to a database that warehouses data for an application. The Examiner has failed to show that the references teach all of the claimed features. The closest the Examiner has come is to suggest receiving data from a database, transforming it to an intermediary format, transforming the intermediary format back to the original format, and transmitting the data to another database (and not to an application).

The claimed invention is not needed for an application that can access the database of Straube

The Examiner is again pointed to page 2, lines 10-18 of the specification. According to the specification, one of the problems in application integration is that applications do not or cannot directly access the distributed directory. If the database of Straube is truly analogous to the distributed directory of the claimed invention, then anything that can access Straube's database is able to access the distributed directory of the claimed invention. But if an application can access that data, then that application does not suffer the failing that the claimed invention was designed to address: namely, the sharing of data affecting the distributed directory when the application cannot access the distributed directory.

For the foregoing reasons, reconsideration and allowance of claims 7, 9, 11, 15, 18-20, 22, 26, 27, and 29-42 of the application as amended is solicited. The Examiner is encouraged to telephone the undersigned at (503) 222-3613 if it appears that an interview would be helpful in advancing the case.

Respectfully submitted,
MARGER JOHNSON & McCOLLOM, P.C.

Ariel S. Rogson
Reg. No. 43,054

MARGER JOHNSON & McCOLLOM, P.C.
1030 SW Morrison Street
Portland, OR 97205
503-222-3613
Customer No. 45842